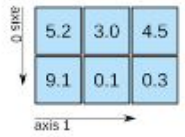
Chapter 3

Python (Data Types Intro)

Data Type VS Data Structure

- Data Types defines the characteristics and behaviour of Individual values that can be stored in data structure
- Example: string, float, integer, boolean

- Data Structure defines the layout and organization of data
- It provides a means to efficiently access, manipulate, and manage the data
- Example: Array, Stack, etc



"Think about storing foods in Refrigerator"

Python Data Types

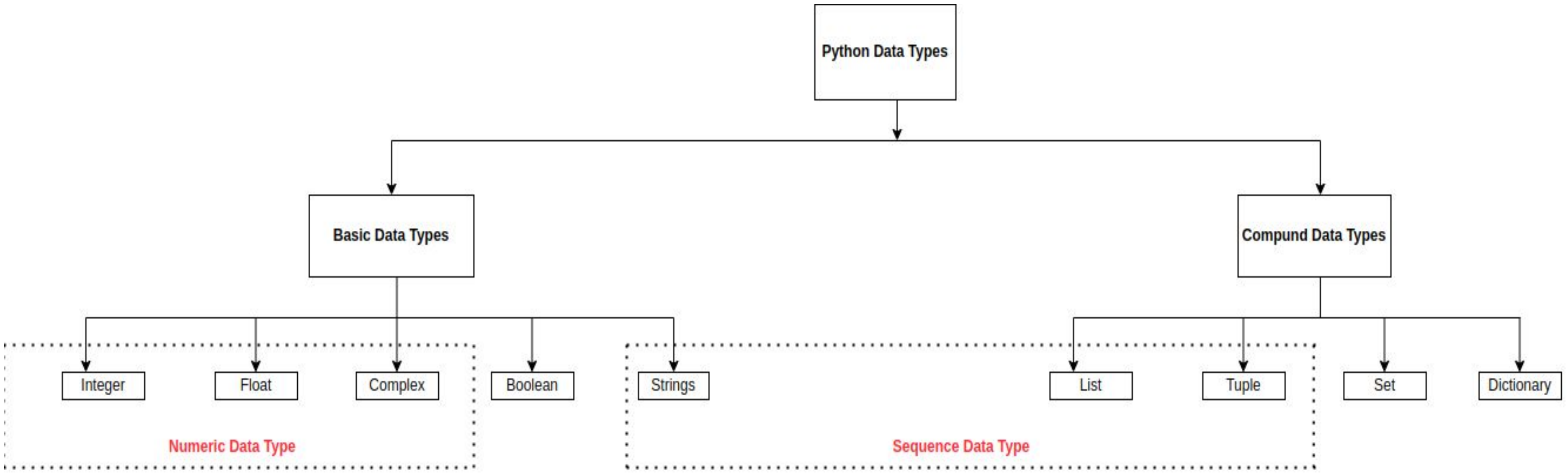
- Data Types refers to the **classification of data** values based on their characteristics.

- Python Data Type means how Python represents different types of data.

- Python Data Types can be categorized in two broad categories:
  - **Basic Data Types**
    - They are fundamental Data Types provided by the language itself.
    - Example: Integer, Float, Complex Numbers, Boolean, String

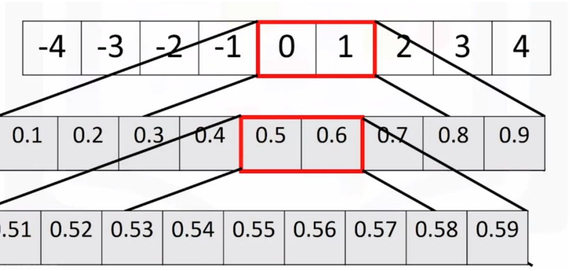
  - **Compound Data Types**
    - They are composed of multiple basic data types or other compound data types.
    - Example: List, Tuple, Set, Dictionary, etc

Python Data Types (Taxonomy)



Basic Data Type (Example)

1. **Integer**
2. **Float**



3. **Complex**

$$\underset{\text{Real part}}{\underset{\uparrow}{a}} + \underset{\text{Imaginary part}}{\underset{\uparrow}{b}}i \longrightarrow \sqrt{-1} \longrightarrow 0 + 1i$$

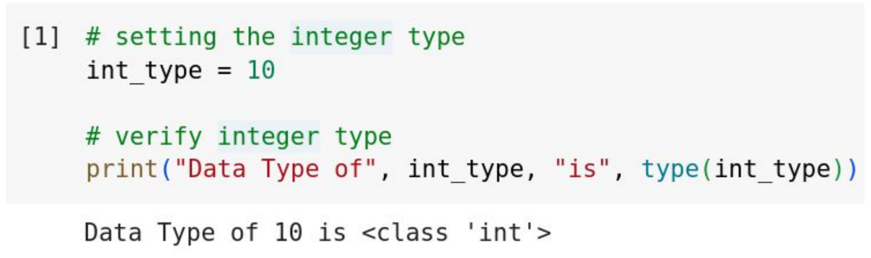
4. **Boolean**

True	**False**

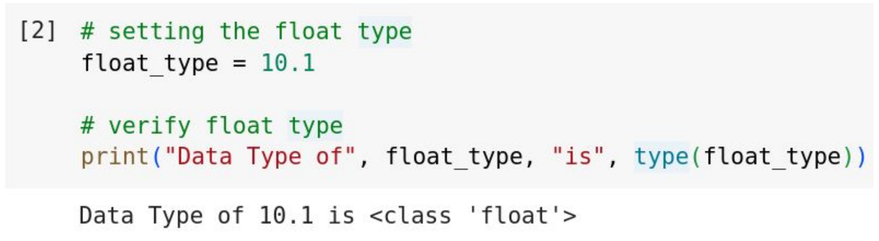
5. **String**

```
"Hello World"
```

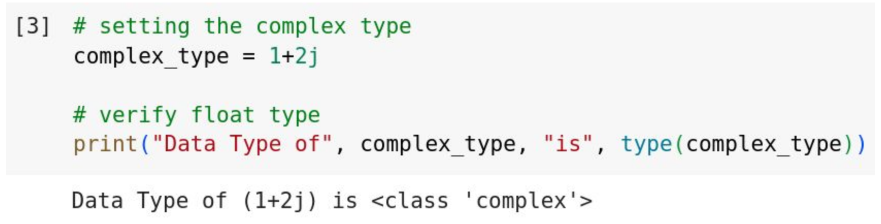
Setting and Getting The Data Type



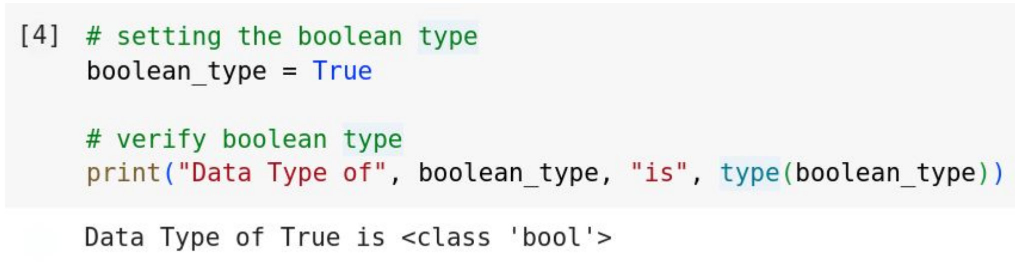
```
[1] # setting the integer type
    int_type = 10

    # verify integer type
    print("Data Type of", int_type, "is", type(int_type))

    Data Type of 10 is <class 'int'>
```



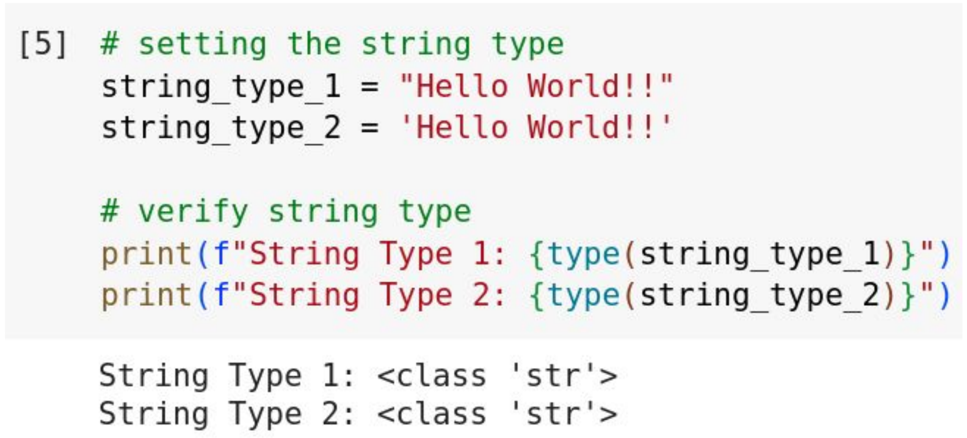
```
[2] # setting the float type
    float_type = 10.1

    # verify float type
    print("Data Type of", float_type, "is", type(float_type))

    Data Type of 10.1 is <class 'float'>
```



```
[3] # setting the complex type
    complex_type = 1+2j

    # verify float type
    print("Data Type of", complex_type, "is", type(complex_type))

    Data Type of (1+2j) is <class 'complex'>
```



```
[4] # setting the boolean type
    boolean_type = True

    # verify boolean type
    print("Data Type of", boolean_type, "is", type(boolean_type))

    Data Type of True is <class 'bool'>
```

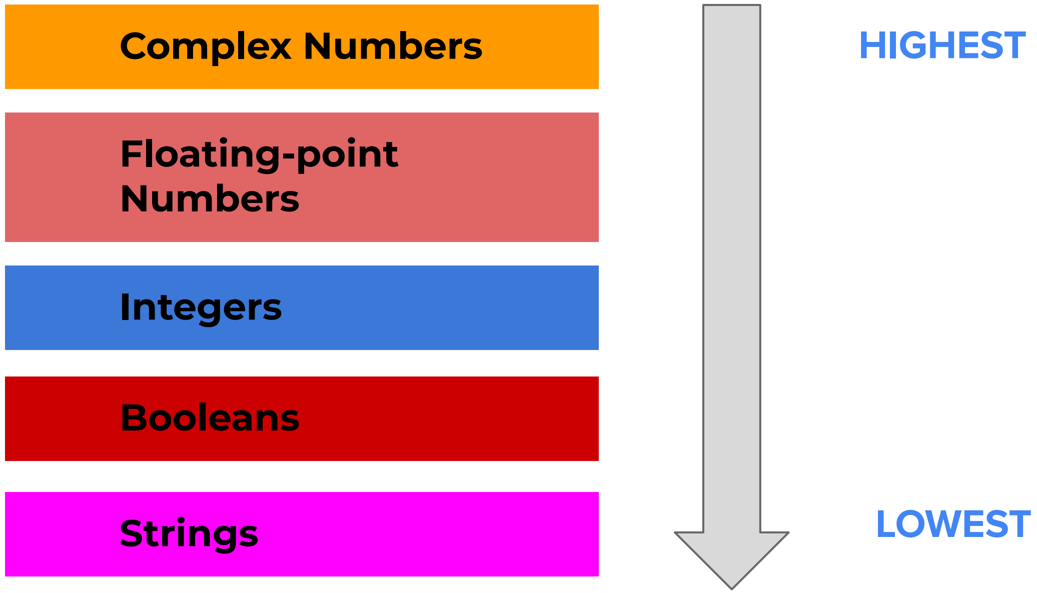


```
[5] # setting the string type
    string_type_1 = "Hello World!!"
    string_type_2 = 'Hello World!!'

    # verify string type
    print(f"String Type 1: {type(string_type_1)}")
    print(f"String Type 2: {type(string_type_2)}")

    String Type 1: <class 'str'>
    String Type 2: <class 'str'>
```

Type Casting / Type Conversion

- **Type Casting** is the process of converting data of one type to another.
    - **Example:** converting int to str
- Two Types:
    1. Implicit Type Casting
    2. Explicit Type Casting

Data Type Precedence



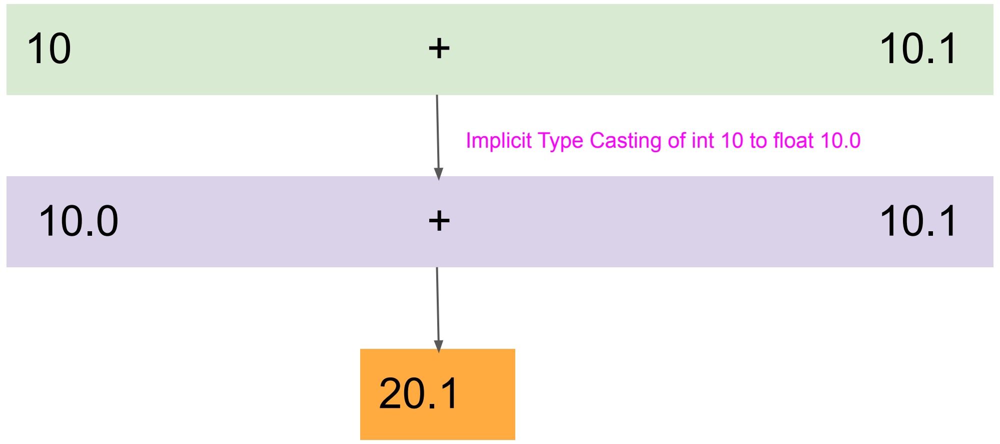
1. Implicit Type Casting

- In this type casting, Python automatically converts one data type to another
- Here, Type conversion of lower data type to higher data type will occur automatically.

Addition of Integer and Float Data Types

```python
int_num = 10
float_num = 10.1
print(int_num + float_num)
```

```
20.1
```



1. Implicit Type Casting (CONT)

Q. Why didn't Python convert the float to an int type and perform the addition with the same type?

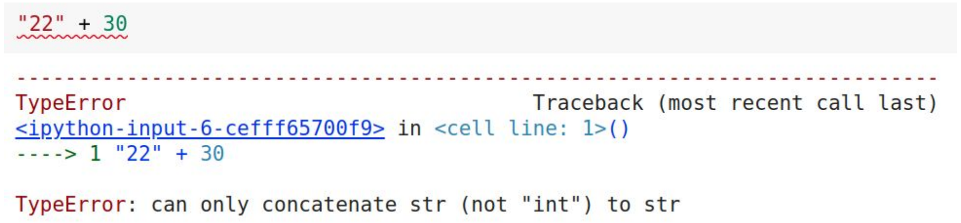
>> Ans:

- Floats have higher precedence than Integers in data type hierarchy

- To Prevent from Data Loss (converting 10.1 to 10 leads to loss in data)

2. Explicit Type Casting

Problem: *Add string type and integer type*

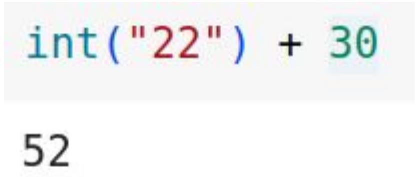


```
"22" + 30
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-6-cefff65700f9> in <cell line: 1>()
----> 1 "22" + 30

TypeError: can only concatenate str (not "int") to str
```

Solution: *Explicit Type Casting*

In Explicit Type Casting, User convert the data type of an object to required data type



```
int("22") + 30
```

```
52
```

Class Work

Q. Develop basic calculator app for the user:

a. Input two numbers from user and assign it to variable num1 and num2
b. Perform addition of two numbers and assign to variable result_add
c. Display result_add to user.

Hint: *use built-in input() function to get user input*

References

- https://cognitiveclass.ai/courses/python-for-data-science
- https://www.dataquest.io/blog/data-structures-in-python/
- https://www.w3schools.com/python/python_datatypes.asp